1. $C(H) = \frac{number\ of\ edges\ in\ G_H}{maximum\ number\ edges\ in\ G_H} = \frac{2m_i}{n_i(n_i-1)} = \frac{2\times2}{4\times3} = \frac{1}{3}$

   m_i: MJ, JK
   n_i: E, M, J, K

2. $e(M) = \max_j\{d(M, v_j)\} = d(M, C) = 1 + 2 + 2 + 1 + 3 = 9$

   The distance between M and other nodes are shorter than 9.

3. $c(H) = \frac{n-1}{\Sigma_j d(H,v_j)} = \frac{16}{6+4+7+3+1+3+4+3+2+5+2+3+6+3+10+4} = \frac{8}{33}$

4. a. Inflation = 1.1

```
[[0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022 0.022]
 [0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045 0.045]
 [0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034 0.034]
 [0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043 0.043]
 [0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057 0.057]
 [0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042 0.042]
 [0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135 0.135]
 [0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048 0.048]
 [0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112 0.112]
 [0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092 0.092]
 [0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041 0.041]
 [0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105 0.105]
 [0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033]
 [0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138 0.138]
 [0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033 0.033]
 [0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018 0.018]
 [0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001]]
```

Inflation = 1.3

```
[[0.0113 0.0111 0.0113 0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.3181 0.3131 0.3176 0.0004 0.0004 0.0004 0.0004 0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.1233 0.1214 0.1232 0.0002 0.0001 0.0001 0.0001 0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.0004 0.0004 0.0004 0.0008 0.0008 0.0008 0.0008 0.0001 0.0001 0.0001 0.0001 0.0001 0.     0.     0.     0.     0.0001]
 [0.0022 0.0022 0.0022 0.004  0.004  0.004  0.004  0.0007 0.0006 0.0006 0.0006 0.0006 0.     0.     0.     0.     0.0006]
 [0.0004 0.0004 0.0004 0.0007 0.0007 0.0007 0.0007 0.0001 0.0001 0.0001 0.0001 0.0001 0.     0.     0.     0.     0.0001]
 [0.5437 0.5507 0.5443 0.9912 0.9911 0.9911 0.9912 0.1656 0.1506 0.1385 0.1445 0.1502 0.0001 0.0001 0.0001 0.0001 0.1502]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.0002 0.0002 0.0002 0.0005 0.0005 0.0005 0.0005 0.0063 0.0064 0.006  0.0062 0.006  0.0001 0.0001 0.0001 0.0001 0.0066]
 [0.0001 0.0001 0.0001 0.0002 0.0002 0.0002 0.0002 0.003  0.003  0.0028 0.003  0.0028 0.0001 0.0001 0.0001 0.0001 0.0031]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.0001 0.0001 0.0001 0.0002 0.0002 0.0002 0.0002 0.003  0.0031 0.0029 0.003  0.0029 0.0001 0.0001 0.0001 0.0001 0.0032]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.0001 0.0001 0.0001 0.0001 0.    ]
 [0.0002 0.0003 0.0002 0.0019 0.002  0.002  0.0019 0.821  0.8359 0.8488 0.8423 0.8371 0.9995 0.9995 0.9995 0.9995 0.8359]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.0001 0.0001 0.0001 0.0001 0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]]
```

Inflation = 1.5

```
[[0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.9847 0.9847 0.9847 0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.0153 0.0153 0.0153 0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     1.     1.     1.     1.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.583  0.583  0.583  0.583  0.583  0.     0.     0.     0.     0.583 ]
 [0.     0.     0.     0.     0.     0.     0.     0.2643 0.2643 0.2643 0.2643 0.2643 0.     0.     0.     0.     0.2643]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.1528 0.1528 0.1528 0.1528 0.1528 0.     0.     0.     0.     0.1528]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     1.     1.     1.     1.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]]
```

Inflation = 1.7

```
[[0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [1.     1.     1.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     1.     1.     1.     1.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.1505 0.152  0.148  0.1522 0.1481 0.     0.     0.     0.     0.1513]
 [0.     0.     0.     0.     0.     0.     0.     0.8254 0.824  0.8278 0.8238 0.8278 0.     0.     0.     0.     0.8247]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.0241 0.0241 0.0242 0.0241 0.0242 0.     0.     0.     0.     0.0241]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     1.     1.     1.     1.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.     0.    ]]
```

Inflation = 2.1

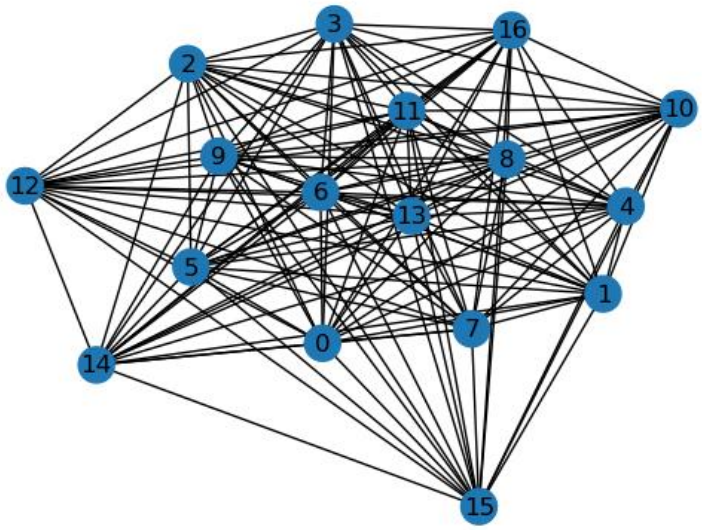
```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

b. The nodes are numbered as following:

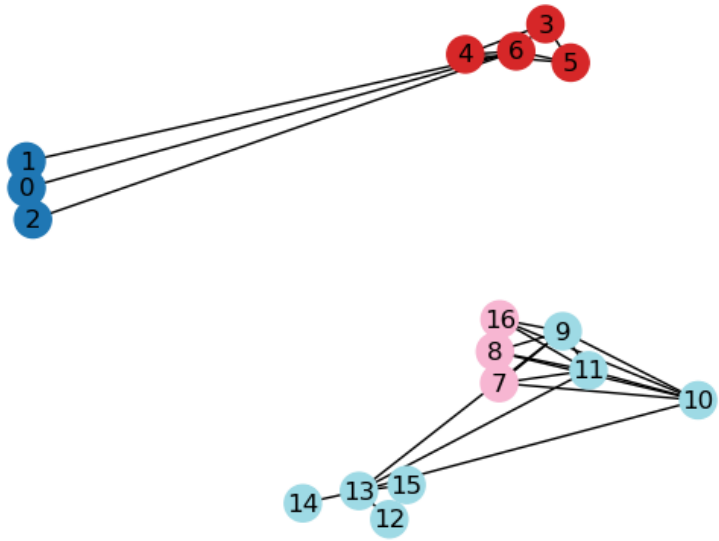
0 to 7: A to H, 8 to 12: J to N, 13 to 16: P to S

For each inflation value, the communities are shown below.

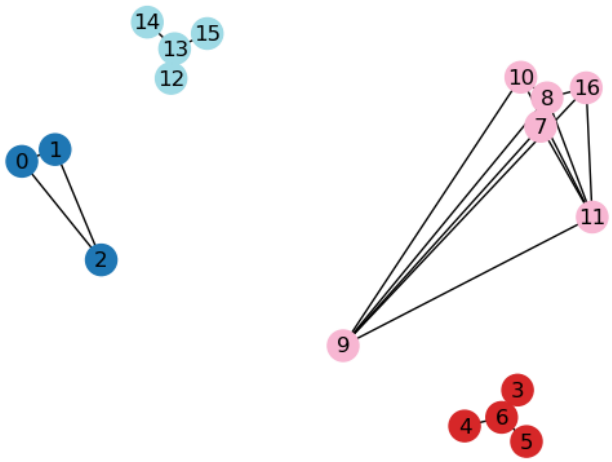
Inflation = 1.1



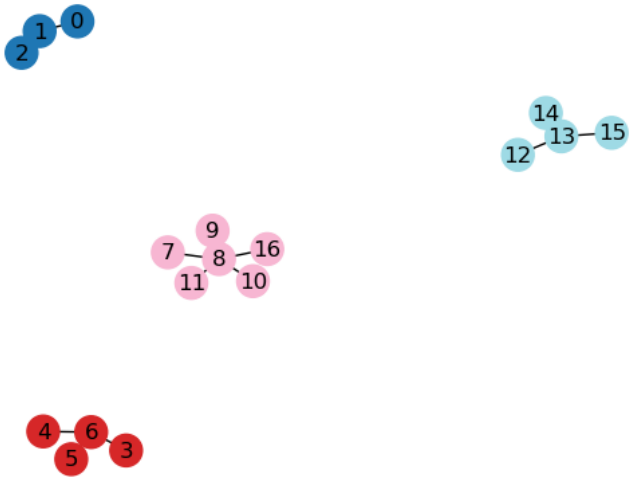
Inflation = 1.3



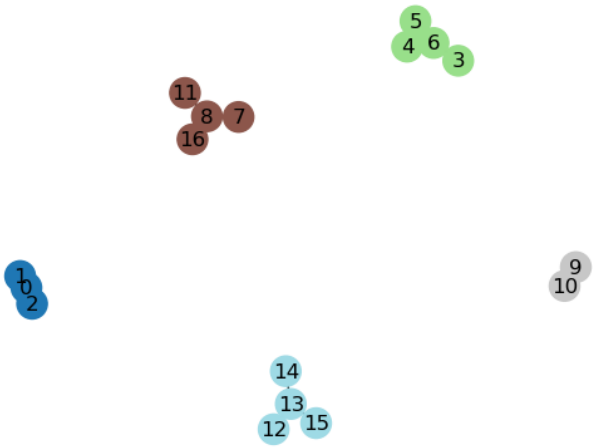
Inflation = 1.5



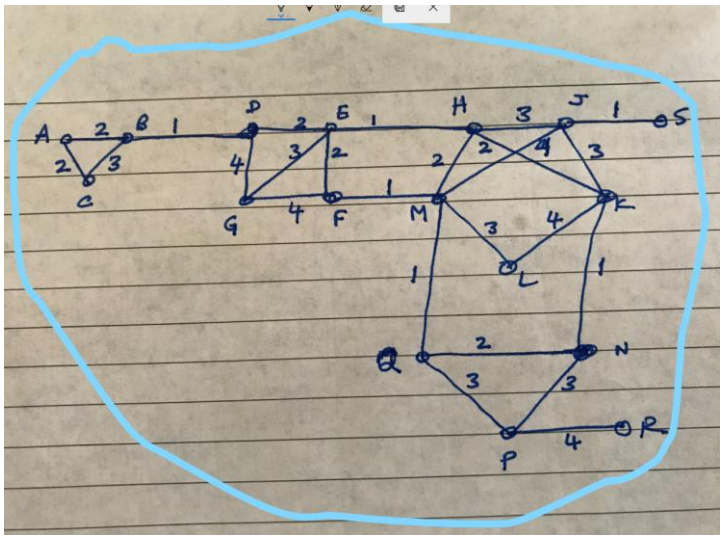
Inflation = 1.7



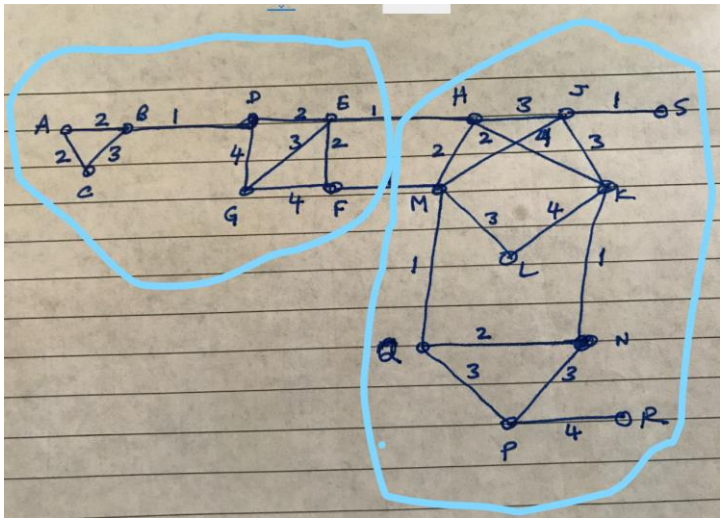
Inflation = 2.1



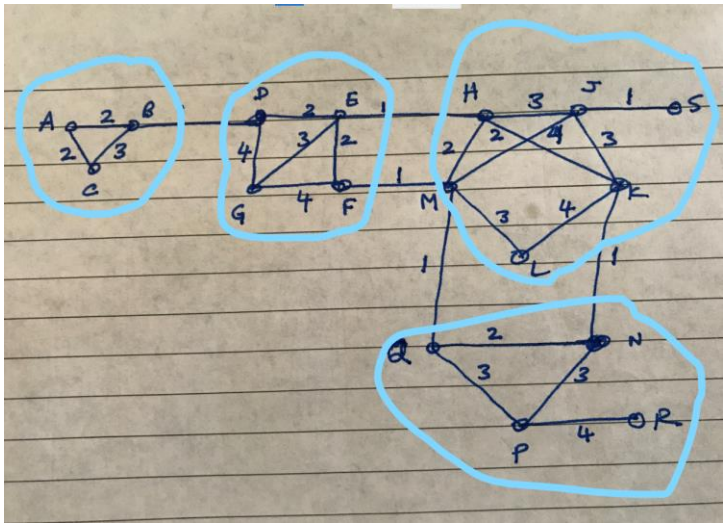
c. Inflation = 1.1



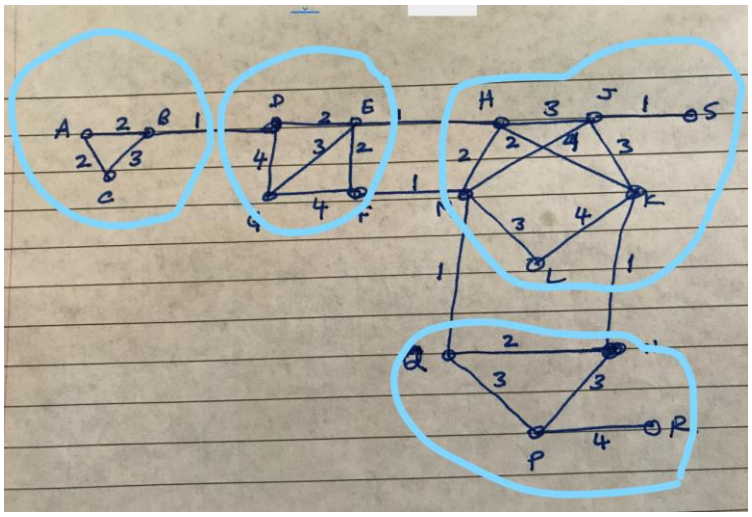
Inflation = 1.3



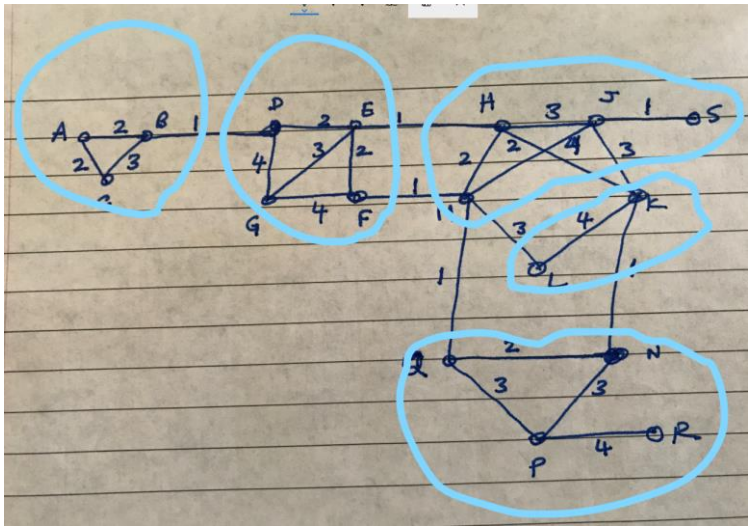
Inflation = 1.5



Inflation = 1.7



Inflation = 2.1



d. Yes. By increasing the inflation rate, we can reduce the size of the communities. I would like to highly connected subgraph to be in a same community. By changing the inflation rate, we can achieve the desired communities.

The python code is attached below.

Code:

```
import numpy as np

data = []

#push edge Eij to the data [vi, vj, |Eij|]
data.append([0, 1, 2])
data.append([0, 2, 2])
data.append([1, 2, 3])
data.append([1, 3, 1])
data.append([3, 4, 2])
data.append([3, 6, 4])
data.append([4, 5, 2])
data.append([4, 6, 3])
data.append([4, 7, 1])
data.append([5, 6, 4])
data.append([5, 11, 1])
data.append([7, 8, 3])
data.append([7, 9, 2])
data.append([7, 11, 2])
data.append([8, 16, 1])
data.append([8, 9, 3])
data.append([8, 11, 4])
data.append([9, 10, 4])
data.append([9, 12, 1])
data.append([10, 11, 3])
data.append([11, 14, 1])
data.append([12, 14, 2])
data.append([12, 13, 3])
data.append([13, 15, 4])
data.append([13, 14, 3])
```

```
#construct adjacency matrix
adjmat = np.zeros((17, 17))
for i in range(17):
    adjmat[i][i] = 1    #diagonal

for edge in data:
    adjmat[edge[0]][edge[1]] = edge[2]
    adjmat[edge[1]][edge[0]] = edge[2]    #symmetric matrix

#normalize adjacency matrix
adj_sum = np.sum(adjmat, axis = 0)

for i in range(17):
    adjmat[:, i] = np.divide(adjmat[:, i], adj_sum[i])

#mcl
infla = [1.1, 1.3, 1.5, 1.7, 2.1]
results = []

for inflation in infla:
    temp = adjmat
    temp_sum = np.zeros(17)
    for iter in range(10):
        temp = np.dot(temp, temp)    #expand
        temp = np.power(temp, inflation)    #inflation
        #normalization
        temp_sum = np.sum(temp, axis = 0)
        for i in range(17):
            temp[:, i] = np.divide(temp[:, i], temp_sum[i])
    results.append(temp)

k = 4
#np.around(results[k], decimals=3)
np.set_printoptions(precision = 2, suppress = True, linewidth=200)
print(results[k])
```

```
import markov_clustering as mcl

final_mat = []
final_cl = []
for i in infla:
    result = mcl.run_mcl(adjmat, inflation = i, iterations = 10)
    final_mat.append(result)
    cluster = mcl.get_clusters(result)
    final_cl.append(cluster)

print(result)
k = 4
mcl.draw_graph(final_mat[k], final_cl[k], with_labels = True, edge_color = "black")
```